GAIN ALLOCATION IN PROPORTIONATE-TYPE NLMS ALGORITHMS FOR FAST DECAY OF OUTPUT ERROR AT ALL TIMES

Kevin T. Wagner

Naval Research Laboratory
Radar Division
Washington, DC 20375, USA
kevin.wagner@nrl.navy.mil

Miloš I. Doroslovački

The George Washington University
Department of Electrical and Computer Engineering
Washington, DC 20052, USA
doroslov@gwu.edu

ABSTRACT

In this paper, we propose three new proportionate-type NLMS algorithms: the water filling algorithm, the feasible water filling algorithm, and the adaptive μ-law proportionate NLMS (MPNLMS) algorithm. The water filling algorithm attempts to choose the optimal gains at each time step. The optimal gains are found by minimizing the mean square error (MSE) at each time with respect to the gains, given the previous mean square weight deviations. While this algorithm offers superior convergence times, it is not feasible. The second algorithm is a feasible version of the water filling algorithm. The adaptive MPNLMS (AMPNLMS) algorithm is a modification of the MPNLMS algorithm. In the MPNLMS algorithm, the parameter μ of the μ-law compression is constant. In the AMPNLMS algorithm the parameter μ is allowed to vary with time. This modification allows the algorithm more flexibility when attempting to minimize the MSE. Compared with several feasible algorithms, the AMPNLMS algorithm has the fastest MSE decay for almost all times.

Index Terms — Adaptive filters, least mean square methods, convergence of numerical methods, sparse impulse response.

1. PTNLMS ALGORITHM FRAMEWORK

Beginning with the proportionate normalized least mean square (PNLMS) algorithm [1], proportionate-type NLMS (PtNLMS) algorithms have been widely used in adaptive filtering applications when dealing with sparse impulse responses. A sparse impulse response is one in which very few coefficients differ from zero.

In this section we lay out the framework of a canonical PtNLMS algorithm. Our general goal is to estimate an unknown system impulse response of length L given by $\mathbf{w} = [w_1, w_2, \ldots, w_L]$. The input signal to the unknown system at time k is $\mathbf{x}(k) = [x(k), x(k-1), \ldots, x(k-L+1)]^T$. The output of the system is $y(k) = \mathbf{w}^T\mathbf{x}(k)$. The measured output of the system, $d(k)$, contains noise $v(k)$ and is equal to the sum of $y(k)$ and $v(k)$. The algorithm is designed to estimate the system impulse response. We define the estimated system impulse response as $\hat{\mathbf{w}}(k)$. The estimated system output is given by $\hat{y}(k) = \hat{\mathbf{w}}^T(k)\mathbf{x}(k)$. We measure the error, defined as $e(k)$, of our estimate by taking the difference between the measured output and the estimated output.

Table I

PtNLMS Algorithm

$y(k)$	$=$	$\mathbf{w}^T\mathbf{x}(k)$
$\hat{y}(k)$	$=$	$\hat{\mathbf{w}}^T\mathbf{x}(k)$
$d(k)$	$=$	$y(k) + v(k)$
$e(k)$	$=$	$d(k) - \hat{y}(k)$
$\hat{\mathbf{w}}(k+1)$	$=$	$\hat{\mathbf{w}}(k) + \frac{\beta\mathbf{G}(k+1)\mathbf{x}(k)e(k)}{\mathbf{x}^T(k)\mathbf{G}(k+1)\mathbf{x}(k)+\delta}$

Table II

PtNLMS Gain Matrix Generation

Specify		$F\left[\lvert\hat{w}_l(k)\rvert\right]$
$\gamma_{\min}(k+1)$	$=$	$\rho\max\{\delta_{\mathrm{p}}, F\left[\lvert\hat{w}_1(k)\rvert\right], \ldots, F\left[\lvert\hat{w}_l(k)\rvert\right]\}$
$\gamma_l(k+1)$	$=$	$\max\{\gamma_{\min}(k+1), F\left[\lvert\hat{w}_l(k)\rvert\right]\}$
$g_l(k+1)$	$=$	$\frac{\gamma_l(k+1)}{\frac{1}{L}\sum_{i=1}^{L}\gamma_i(k+1)}$
$\mathbf{G}(k+1)$	$=$	$\mathbf{diag}\{g_1(k+1), \ldots, g_L(k+1)\}$

The update equation for the canonical PtNLMS algorithm is given in Table I. The δ term is introduced into the denominator to avoid division by zero in situations where the inputs signal has no energy. In Table II we specify the generation of the time varying gain matrix. The user starts by choosing an appropriate function of the estimated impulse response $F[|\hat{w}_l(k)|]$, $l = 1, 2, \cdots, L$. For instance if we specify $F[|\hat{w}_l(k)|] = 1$ the algorithm becomes the well known NLMS algorithm. Next a minimum gain $\gamma_{\min}(k+1)$ is calculated. The minimum gain ensures that all coefficients are updated at each iteration. The constant δ_{p} is important in the beginning of the algorithm when all of the coefficients are zero and together with ρ prevents the very small coefficients from stalling.

2. RECURSIVE WEIGHT DEVIATION EQUATIONS

The majority of the analysis that follows is based on the recursion equations for the weight deviation and the square weight deviation. In this section we present these recursion equations in vector and component-wise form. The weight deviation at time k is defined as $\mathbf{z}(k) = \mathbf{w} - \hat{\mathbf{w}}(k)$. The update equation for the weight deviation vector is

$$
\begin{aligned}
\mathbf{z}(k+1) \quad &= \quad \mathbf{z}(k) - \frac{\beta\mathbf{G}(k+1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{z}(k)}{\mathbf{x}(k)\mathbf{G}(k+1)\mathbf{x}(k)+\delta} \\
&- \quad \frac{\beta\mathbf{G}(k+1)\mathbf{x}(k)v(k)}{\mathbf{x}^T(k)\mathbf{G}(k+1)\mathbf{x}(k)+\delta}. \qquad (1)
\end{aligned}
$$

Employing the convention that $x_i(k)$ is the i^{th} component of vector $\mathbf{x}(k)$, we can express (1) and the square weight deviation $[z_i^2(k+1)]$ in component-wise form as follows:

978-1-4244-2354-5/09/$25.00 ©2009 IEEE

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **MAY 2010**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
Gain Allocation In Proportionate-Type Nlms Algorithms For Fast Decay Of Output Error At All Times

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Research Laboratory Radar Division Washington, DC 20375, USA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002314. IEEE International Conference on Acoustics, Speech and Signal Processing (34th) held in Taipei, Taiwan on 19-24 April 2009. U.S. Government or Federal Purpose Rights License

14. ABSTRACT
In this paper, we propose three new proportionate-type NLMS algorithms: the water filling algorithm, the feasible water filling algorithm, and the adaptive ì-law proportionate NLMS (MPNLMS) algorithm. The water filling algorithm attempts to choose the optimal gains at each time step. The optimal gains are found by minimizing the mean square error (MSE) at each time with respect to the gains, given the previous mean square weight deviations. While this algorithm offers superior convergence times, it is not feasible. The second algorithm is a feasible version of the water filling algorithm. The adaptive MPNLMS (AMPNLMS) algorithm is a modification of the MPNLMS algorithm. In the MPNLMS algorithm, the parameter ì of the ì-law compression is constant. In the AMPNLMS algorithm the parameter ì is allowed to vary with time. This modification allows the algorithm more flexibility when attempting to minimize the MSE. Compared with several feasible algorithms, the AMPNLMS algorithm has the fastest MSE decay for almost all times.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
unclassified	**unclassified**	**unclassified**	**SAR**	**4**	

$$
\begin{aligned}
z_i(k+1) &= z_i(k) - \frac{\beta g_i(k+1)x_i(k)\sum_{j=1}^{L} x_j(k)z_j(k)}{\sum_{j=1}^{L} x_j^2(k)g_j(k+1)+\delta} \\
&- \frac{\beta g_i(k+1)x_i(k)v(k)}{\sum_{j=1}^{L} x_j^2(k)g_j(k+1)+\delta} \qquad (2)
\end{aligned}
$$

$$
\begin{aligned}
z_i^2(k+1) &= z_i^2(k) \\
&+ \frac{\beta^2 g_i^2(k+1)x_i^2(k)\sum_{j=1}^{L}\sum_{l=1}^{L} x_j(k)x_l(k)z_j(k)z_l(k)}{\left[\sum_{j=1}^{L} x_j^2(k)g_j(k+1)+\delta\right]^2} \\
&+ \frac{\beta^2 g_i^2(k+1)x_i^2(k)v^2(k)}{(\sum_{j=1}^{L} x_j^2(k)g_j(k+1)+\delta)^2} \\
&- \frac{2\beta g_i(k+1)x_i(k)\sum_{j=1}^{L} x_j(k)z_j(k)z_i(k)}{\sum_{j=1}^{L} x_j^2(k)g_j(k+1)+\delta} \\
&- \frac{2\beta g_i(k+1)x_i(k)v(k)z_i(k)}{\sum_{j=1}^{L} x_j^2(k)g_j(k+1)+\delta} \\
&+ \frac{2\beta^2 g_i^2(k+1)x_i^2(k)\sum_{j=1}^{L} x_j(k)z_j(k)v(k)}{\left[\sum_{j=1}^{L} x_j^2(k)g_j(k+1)+\delta\right]^2}. \qquad (3)
\end{aligned}
$$

3. OPTIMAL PTNLMS ALGORITHM

We begin by making the following assumptions.

Assumption I: The input signal is a white Gaussian noise process with zero-mean and variance σ_x^2.

Assumption II: β is sufficiently small such that $z_i(k)$ fluctuates much slower than $\mathbf{x}(k)$ and therefore they can be considered independent.

Assumption III: The denominator terms are assumed to be constant. (The validity of this assumption was discussed in [2].)

Assumption IV: The measurement noise $v(k)$ is white with zero-mean, variance σ_v^2, and it is independent of the input.

Under these assumptions and assuming that $g_i(k+1)$ is a deterministic function of time, we arrive at the following equations after taking the expectation of the weight deviation and the square weight deviation:

$$
\begin{aligned}
E\{z_i(k+1)\} &\equiv \overline{z}_i(k+1) = \overline{z}_i(k) - \beta_0 g_i(k+1)\overline{z_i}(k) \qquad (4) \\
E\{z_i^2(k+1)\} &\equiv \overline{z_i^2}(k+1) = \overline{z_i^2}(k) - 2\beta_0 g_i(k+1)\overline{z_i^2}(k) \\
&+ \beta_0^2 g_i^2(k+1)\left[3\overline{z_i^2}(k) + \sum_{j\neq i}\overline{z_j^2}(k)\right] \\
&+ \beta_0^2\frac{\sigma_v^2}{\sigma_x^2}g_i^2(k+1). \qquad (5)
\end{aligned}
$$

Here, we have defined $\beta_0 = \beta\sigma_x^2/(L\sigma_x^2+\delta)$. We make the following definitions for notational convenience:

$$
\begin{aligned}
c_i &= -2\beta_0\overline{z_i^2}(k) \\
q_i &= 2\beta_0^2\left[\left(3\overline{z_i^2}(k) + \sum_{j\neq i}\overline{z_j^2}(k)\right) + \frac{\sigma_v^2}{\sigma_x^2}\right]. \qquad (6)
\end{aligned}
$$

At this point we try to find the optimal gain by minimizing, $J(k+1) = \sigma_v^2 + \sigma_x^2\sum_{i=1}^{L}\overline{z_i^2}(k+1)$ for all k, with respect to the gain and the constraint $\sum_{j=1}^{L} g_i(k+1) = L$. Note that $J(k)$ is the approximate MSE at instant k [3]. We can recast this problem as an optimization problem in the form

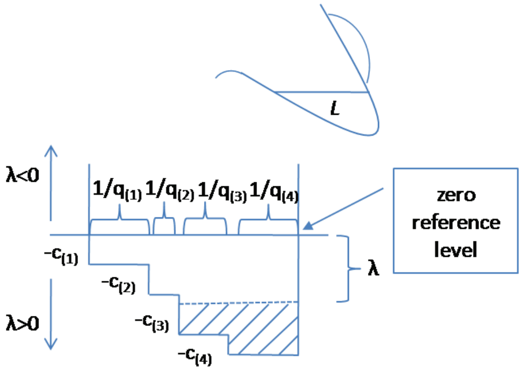

Fig. 1. Water filling solution for λ.

$$
\begin{aligned}
\min_{\mathbf{g}} J(k+1) = \mathbf{c}^T\mathbf{g} + \frac{1}{2}\mathbf{g}^T\mathbf{Q}\mathbf{g} + \sigma_v^2 \\
g_i \geq 0 \; \forall \; i \\
\mathbf{1}^T\mathbf{g} = L \qquad (7)
\end{aligned}
$$

where $\mathbf{g} = [g_1(k+1), g_2(k+1), \ldots, g_L(k+1)]^T$ and $\mathbf{Q} = \mathbf{diag}(q_1, q_2, \ldots, q_L)$.

Next, for convenience, we make the substitution $g_i(k+1) = s_i^2$ and incorporate the constraint into the minimization problem. This results in

$$
T(\mathbf{s}, \lambda) = \sum_{i=1}^{L} c_i s_i^2 + \frac{1}{2}\sum_{i=1}^{L} q_i s_i^4 + \lambda(\sum_{i=1}^{L} s_i^2 - L). \qquad (8)
$$

Taking the first derivative and setting it equal to zero yields two solutions, $s_k = 0$ and $s_k = \sqrt{(-c_k - \lambda)/q_k}$. If we examine the second derivative $\frac{\partial^2 T}{\partial^2 s_k}$ we find that the first solution results in a minimum if $\lambda \geq -c_k$ and the second solution results in a minimum if $\lambda < -c_k$. As it turns out, the solution to this problem is of the "water filling" variety. We choose the constant λ according to the following rules. First we sort the entries of $-\mathbf{c}$ in ascending order to form a new vector such that $-c_{(1)} < -c_{(2)} < \ldots < -c_{(L)}$. We subsequently rearrange the elements of $\mathbf{Q}$ to match the position of the original indices in the sorted $-\mathbf{c}$ and to form a new matrix whose diagonal elements are $q_{(1)}, q_{(2)} \ldots, q_{(L)}$.

Next we solve for

$$
\lambda_i = \frac{-\sum_{k=i}^{L}\frac{c_{(k)}}{q_{(k)}} - L}{\sum_{k=i}^{L}\frac{1}{q_{(k)}}}. \qquad (9)
$$

We choose $\lambda = \lambda_i$ if $-c_{(i-1)} < \lambda_i < -c_{(i)}$, where $-c_{(0)} = -\infty$. We depict this water-filling problem in Fig. 1. We have a pitcher containing a volume L of water. After pouring the contents of the pitcher in a vessel with the profile shown in Fig. 1, we want to know the difference of the height the water reaches in the vessel and the zero-reference level of the vessel. This difference gives us the value of λ we are seeking. The stairway sloped bottom of the vessel is defined by the treads of heights $-c_{(k)}$ and with widths, $1/q_{(k)}$.

4. FEASIBLE SUBOPTIMAL PTNLMS ALGORITHM

The algorithm proposed in the previous sections is not feasible. Specifically we need to know $E\{z_i^2(k)\} = E\{[w_i - \hat{w}_i(k)]^2\}$, which requires knowledge of the optimal impulse response. In order to avoid this we rewrite the error in another form

$$e(k) = \sum_{j=1}^{L} x_j(k) z_j(k) + v(k). \tag{10}$$

Next we multiply both sides of the equation by $x_i(k)$ and take the expectation [assuming $x_i(k)$ is a white signal]. This results in

$$E\{x_i(k)e(k)\} = \sigma_x^2 E\{z_i(k)\}. \tag{11}$$

If we define $p_i(k) = x_i(k)e(k)$, we can calculate $\overline{p}_i(k) = E\{p_i(k)\}$ and then solve for $\bar{z}_i(k)$. We update our estimate of $\overline{p}_i(k)$ in the following fashion:

$$\begin{aligned} \hat{\overline{p}}_i(k) &= \alpha \hat{\overline{p}}_i(k-1) + (1-\alpha) p_i(k) \\ \hat{\bar{z}}_i(k) &= \frac{\hat{\overline{p}}_i(k)}{\sigma_x^2} \end{aligned} \tag{12}$$

where $0 \leq \alpha \leq 1$. Now, we approximate $E\{z_i^2(k)\}$ with $\left[\hat{\bar{z}}_i(k)\right]^2$. This approximation is expected to be good at the beginning when $z_i(k)$ are large. The same approximation is used in [4], [5]. Better estimates of $E\{z_i^2(k)\}$ are possible, but more complex to compute.

5. ADAPTIVE MPNLMS ALGORITHM

The AMPNLMS algorithm is a modification of the MPNLMS algorithm [6], [7]. As in the previous section we try to reach the optimal performance as quickly as possible at each time step. In order to do so we introduce a time varying $\epsilon_c(k)$, such that $F[|\hat{w}_l(k)|] = \ln[1 + \mu(k)|\hat{w}_l(k)|]$ and $\epsilon_c(k) = 1/\mu(k)$. We recall that ϵ_c is related to when a coefficient is considered to be converged [7]. The strategy employed by the AMPNLMS is to start out with a large value for $\epsilon_c(k)$ and slowly decrease the required ϵ-neighborhood to be reached by the converged algorithm. In doing so, the AMPNLMS algorithm initially behaves like the PNLMS algorithm and then transitions to performing like the NLMS algorithm as time proceeds. This transition reflects our knowledge of the impulse response. Initially we know the impulse response is sparse and therefore we direct our resources in a manner to most efficiently estimate the impulse response. As time goes on, our *a priori* knowledge [which is $\hat{\mathbf{w}}(k)$] becomes close to the true value of the impulse response and the coefficient estimate errors are uniformly distributed along all coefficients, in contrast to the initial situation.

We determine the value we choose for $\epsilon_c(k)$ by relating it to the current MSE $\overline{e^2}(k)$. We can form an estimate of $\overline{e^2}(k)$ and update $\mu(k)$ by calculating:

$$\begin{aligned} \zeta(k+1) &= \xi \zeta(k) + (1-\xi) e^2(k) \\ \tilde{\epsilon}_L(k+1) &= \frac{\zeta(k+1)}{\nu} \\ \epsilon_c(k+1) &= \sqrt{\frac{\tilde{\epsilon}_L(k+1)}{L\sigma_x^2}} \\ \mu(k+1) &= \frac{1}{\epsilon_c(k+1)} \end{aligned} \tag{13}$$

where $0 \leq \xi \leq 1$ and ν is a constant relating the current MSE estimate (obtained by time averaging) to $\tilde{\epsilon}_L(k+1)$. The term $\zeta(k+1)$ is an estimate of $\overline{e^2}(k+1)$. $\tilde{\epsilon}_L(k+1)$ is the distance to the steady-state MSE ("noise floor") that is considered as achievement of convergence when reached by the MSE.

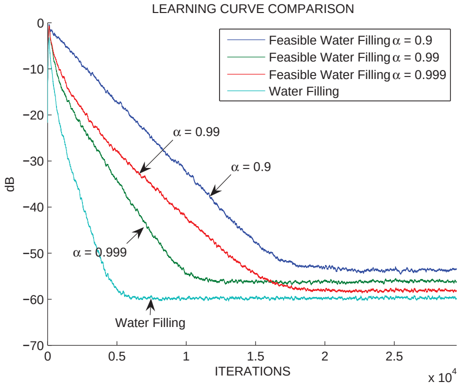

Fig. 2. Water filling α comparison.

In essence, the algorithm continuously tightens the level above the noise floor that the algorithm's MSE is required to achieve. Proceeding in this fashion causes the algorithm to behave like the PNLMS algorithm initially when changing large coefficients results in large changes in the MSE. As time proceeds and the MSE bound tightens, coefficients with smaller amplitudes are adjusted until the algorithm finally performs like the NLMS (through the μ-compression law). Adaptation of $\mu(k)$ is done by using the estimate of MSE, $\zeta(k)$. In this way, the changes of $\mu(k)$ are related in a natural way to the changes of MSE, i.e., when the MSE becomes smaller, the algorithm inclines to treat all coefficients more equally.

6. RESULTS

The parameters used to generate the curves that follow are: $L = 512$, $\beta = 0.1$, $\rho = 0.01$, $\delta = 5.12$, $\sigma_x^2 = 1$, and $\sigma_v^2 = 10^{-6}$. The impulse response used in simulations corresponds to a real-world network echo path similar to the one given in [6], [7]. In Fig. 2 we plot the MSE estimate obtained by Monte Carlo simulations for the non-feasible water filling algorithm and the feasible water filling algorithm with values of α = 0.9, 0.99, and 0.999. As expected, the non-feasible water filling algorithm performs best. The feasible water filling algorithm converges fastest for $\alpha = 0.99$.

We compare the estimated MSE performance of the AMPNLMS using different values of ν in Fig. 3. Specifically, we have chosen $\nu = 1, 10, 100,$ and 1000. The value of $\xi = 0.99$ is fixed in these simulations. Out of four considered values for ν, the value ν=1000 provides the algorithm with the best performance.

Next in Fig. 4, we compare the estimated MSE performance of the AMPNLMS algorithm for a fixed value of $\nu = 1000$, and varying ξ = 0.5, 0.9, 0.99, and 0.999. Varying the value of ξ does not have as great an effect on the MSE performance as the ν parameter does.

In Fig. 5 we compare the NLMS, PNLMS, MPNLMS, ϵ-PNLMS [8], AMPNLMS, feasible water filling, and water filling algorithms. The additional parameters used to generate these curves are as follows: $\alpha = 0.99$, $\xi = 0.99$, and $\nu = 1000$. Naturally, the non-feasible water-filling algorithm performs better than all of the other algorithms. The AMPNLMS algorithm has the second best perfor-

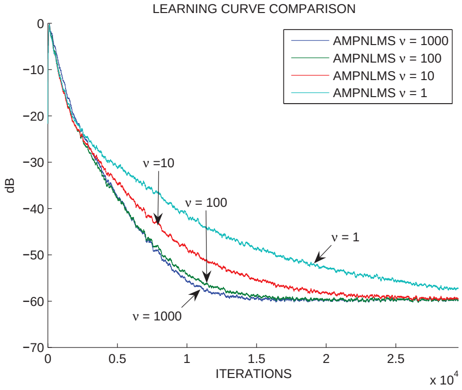

Fig. 3. AMPNLMS ν comparison.

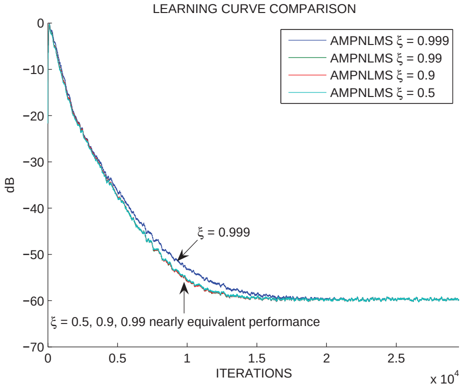

Fig. 4. AMPNLMS ξ comparison.

mance, however; the feasible water-filling algorithm outperforms the AMPNLMS in the initial convergence stage. The AMPNLMS and feasible water-filling algorithms are sensitive to the choice of ξ and α parameters.

Note the appearance of a significant excess MSE in the case of the feasible water filing algorithm. This originates from (a) the inadequacy of (12) for approximating $E\{z_i^2(k)\}$ when the deviations are small, and (b) fluctuation of estimates.

The feasible water filling algorithm has a very high computational complexity per input sampling period. The increased computational complexity is due to the sort operation as well as the need to compute the $\mathbf{c}$ and $\mathbf{Q}$ elements. In contrast, the AMPNLMS requires five multiplications, one addition, and a square root operation (which maybe approximated) beyond the MPNLMS algorithm [7].

7. CONCLUSIONS

We have proposed two new feasible PtNLMS algorithms. The feasible water filling algorithm was motivated by trying to find the optimal gain at each step. This approach resulted initially in the non-feasible water filling algorithm. We subsequently modified the non-feasible version into a suboptimal feasible version.

Next we introduced the AMPNLMS algorithm. This algorithm was based on allowing μ-law coefficient compression to be time-varying. In doing so, the ϵ-neighboorhood, which the algorithm is required to converge to, is slowly tightened as time goes by. The tightening is controlled by the evolving MSE. This modification results in superior MSE convergence.

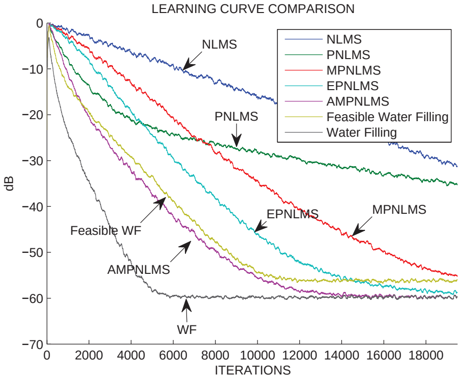

Fig. 5. Algorithm comparison.

8. REFERENCES

[1] D. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancellers," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 508-518, Sept. 2000.

[2] M. Doroslovački and H. Deng, "On convergence of proportionate-type NLMS adaptive algorithms," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006, vol. 3, pp. 105-108.

[3] K. Wagner and M. Doroslovački, "Towards analytical convergence analysis of proportinate-type NLMS algorithms," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, Nevada, March 30-April 4, 2008, pp. 3825-3828.

[4] N. Li, Y. Zhang, Y. Hao, and J. A. Chambers, "A new variable step-size NLMS algorithm designed for applications with exponential decay impulse responses," *Signal Processing*, vol. 88, no.9, pp. 2346-2349, Sept. 2008.

[5] H.-C. Shin, A. H. Sayed, W.-J. Song, "Variable step-size NLMS and affine projection algorithms," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 132-135, Feb. 2004.

[6] H. Deng and M. Doroslovački, "Improving convergence of the PNLMS algorithm for sparse impulse response identification," *IEEE Signal Process. Lett.*, vol. 12, no. 3, pp. 181-184, Mar. 2005.

[7] H. Deng and M. Doroslovački, "Proportionate adaptive algorithms for network echo cancellation," *IEEE Trans. Signal Processing*, vol. 54, no. 5, pp. 1794-1803, May 2006.

[8] K. Wagner, M. Doroslovački, and H. Deng "Convergecne of proportionate-type NLMS adaptive filters and choice of gain matrix," in *Proc. 40th Asilomar on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 29 - Nov. 1, 2006, pp. 243-247.